

$Exp ::= Bool \mid Float \mid Int \mid VarId$ $ Exp + Exp \ [strict]$ $ Exp * Exp \ [strict]$ $ Exp / Exp \ [strict]$ $ Exp \leq Exp \ [seqstrict]$ $ not \ Exp \ [strict]$ $ Exp \ and \ Exp \ [strict(1)]$ $ Stmt ; Exp$ $ ++ \ VarId$ $ \& \ VarId$ $ malloc \ Exp \ [strict]$ $ * \ Exp \ [strict]$ $ \lambda List^{()}\{VarId\} . Exp$ $ \mu VarId . Exp$ $ Exp \ List^{()}\{Exp\} \ [strict]$ $ callcc \ Exp \ [strict]$ $ randomBool$ $ new-agent \ Stmt$ $ me$ $ parent$ $ receive$ $ receive-from \ Exp \ [strict]$ $ quote \ Exp$ $ unquote \ Exp$ $ eval \ Exp \ [strict]$	$Stmt ::=$ $ Stmt ; Stmt$ $ \{ vars \ List\{VarId\} ; Stmt \}$ $ if \ Exp \ then \ Stmt \ else \ Stmt \ [strict(1)]$ $ while \ Exp \ do \ Stmt$ $ output \ Exp \ [strict]$ $ Exp := Exp \ [strict(2)]$ $ aspect \ Stmt$ $ spawn \ Stmt$ $ acquire \ Exp \ [strict]$ $ free \ Exp \ [strict]$ $ release \ Exp \ [strict]$ $ rv \ Exp \ [strict]$ $ send-asynch \ Exp \ Exp \ [strict]$ $ send-synch \ Exp \ Exp \ [strict]$ $ halt-thread$ $ halt-agent$ $ halt-system$
---	---

Figure 5.14: Syntax of the CHALLENGE language